Chapter 8

UTILIZING HYBRID GENETIC ALGORITHMS

Jeffrey A. Joines and Michael G. Kay

Abstract Genetic algorithms (GAs) have been shown to be quite effective at solving a wide range of difficult problems. They are very efficient at exploring the entire search space; however, they are relatively poor at finding the precise local optimal solution in the region in which the algorithm converges. Hybrid GAs are the combination of local improvement procedures, which are good at finding local optima, and genetic algorithms. Hybrid GAs have been shown to be quite effective at solving a wide range of problems. How the GA (the global explorer) and the local improvement procedure (the local exploiter) are combined is extremely important with respect to the final solution quality as well as the computational efficiency of the algorithm. Several different combination strategies will be investigated to determine the most effective method. Furthermore, a new adaptive memory technique will be used to enhance these methods.

Keywords: Hybrid GAs, Lamarckian Evolution, Baldwin Effect, Random Linkage, Global Optimization, Clustering Methods

1. Introduction

There exists a great deal of literature on various optimization techniques. In general, it is extremely difficult to locate the optimal set of values for a system. However, for systems with special structures, e.g., convex of linear functions, etc., optimization may be relatively easy. For example, consider the optimization of the continuous, unbounded univariate function, \( y = -x^2 \). Simple calculus methods provide a simple solution finding the maximum of such simple functions (i.e., set the first derivative equal to zero and solve for \( x \) and then use the second derivati-
tive at that point to determine if it is a maximum, minimum, or an inflection point). For our simple function, \( \frac{dy}{dx} = -2x = 0 \) when \( x = 0 \) and the second derivative is always negative thus indicating \( x = 0 \) is a maximum. However, not all equations yield to such simple solutions to solve equations, e.g., \( y = x \cos(x) \), \( \frac{dy}{dx} = \cos(x) - x \sin(x) = 0 \), which of course has an infinite number of solutions (and can not be solved).

The problem of unsolvable equations led to the development of numerical methods for searching for solutions to equations, i.e., searching for the value(s) of \( x \) that yields \( \frac{dy}{dx} = 0 \). A great number of these search methods have been developed, e.g., gradient search, conjugate gradient search, Newton-Raphson search, etc. However, all of these search routines only find a single root to the equation. When the equation has multiple roots, as \( y = x \cos(x) \) does, the search will only find one local extreme point, not necessarily the globally optimal point. For example, \( y = x + 10 \sin(5x) + 7 \cos(4x) \) is a simple multi-modal uni-variate problem seen in Figure 8.1. If a hill-climbing technique is started at \( x = 2 \) or \( x = 7 \), the technique will climb to the top of only the current hill which represents two local maxima as seen in Figure 8.1. Obvious other starting points could be used to find the global optimal for such a simple function. This multi-start procedure will discussed later.

![Figure 8.1. Simple Uni-variate Function](image)

If the variables of interest are discrete (i.e., variables can only take on whole numbers), the searching process becomes even harder. Also, if the problem does not have gradients, these previous methods will not work. Exact algorithms are guaranteed to find the optimal solution (e.g., simplex algorithm for linear continuous programming problems while branch and bound/cut algorithms can be used for linear integer